IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: David E. MCDYSAN et al.	
Application No.: 09/723,480	Group Art Unit: 2155
Filed: November 28, 2000	Examiner: Bates, K.
Customer No.: 25537 Attorney Docket: RIC00044 Client Docket: 09710_1234	

For: MESSAGE, CONTROL AND REPORTING INTERFACE FOR A DISTRIBUTED NETWORK ACCESS SYSTEM

APPEAL BRIEF

Commissioner for Patents
Alexandria, VA 22313-1450

Dear Sir:

This Appeal Brief is submitted in support of the Notice of Appeal dated June 5, 2006.

I. REAL PARTY IN INTEREST

Verizon is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

III. STATUS OF THE CLAIMS

Claims 1-40 are pending in this appeal, in which no claims have been canceled. No claim is allowed. This appeal is therefore taken from the final rejection of claims 1-40 on March 3, 2006.

IV. STATUS OF AMENDMENTS

The amendment to claims 2, 4-8, 10-12, 14, 17 and 20 filed May 1, 2006 has been entered. No further amendment to the claims has been filed.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed invention addresses problems associated with a network access system. More particularly, the claimed invention relates to an IP-based communication network including a network access system having distributed and separate routing, signaling, service control, filtering, policy control and other functionality from IP forwarding. (Specification, page 1, lines 25-29)

Conventional monolithic router designs have limited flexibility and extensibility. The claimed invention recognizes that it would be desirable, in view of the rapid growth of Internet traffic, to dynamically provision, configure, and/or reallocate access capacity to IP-based services. Because access capacity is necessarily limited and providing additional access capacity is a major cost component of networks, the enforcement of intelligent admission control policies and provision of differing qualities of service is vital to the efficient utilization of available access capacity. However, conventional edge routers are not capable of classifying a wide variety of

traffic types while enforcing policy controls or of responding to dynamic requests for capacity, and this functionality is difficult to incorporate within currently deployed monolithic edge routers. The claimed invention accordingly recognizes that it would be desirable to provide the above as well as additional policy control, network monitoring, diagnostic, and security services in commercialized hardware, while permitting these services to be tailored to meet the needs of individual customers and service providers. (Specification, page 3, line 30 - page 4, line 14)

A distributed network access system architecture including at least an external processor and a programmable access device is introduced. The network access system may further include an access router coupled to the programmable access device.

The external processor transmits a control message to the programmable access device to establish a configuration of the programmable access device. The programmable access device then communicates messages to the external processor for service processing in accordance with the configuration. For example, the control message may be a filter control message that establishes a configuration of a packet header filter in the programmable access device. The packet header filter then communicates network messages filtered from a packet flow in accordance with the configuration established by the control message. To limit communication of network messages from the programmable access device to the external processor, the programmable access device can send a message setting message interface flags in the programmable access device. The external processor may also transmit a monitor control message to the programmable access device to establish a configuration of a monitor in the programmable access device. The programmable access device then communicates reporting messages to the external processor in response to the configuration of the monitor.

Thus, conventional, proprietary edge routers are replaced with a distributed network access system that allocates the functionality of traditional edge routers (as well as additional functionality) among three logical modules: a programmable access device, an external processor, and an access router. Basic routing of packets between input and output ports of the access network is performed by the access router. However, forwarding and generic traffic conditioning functions, such as marking, policing, monitoring, shaping, and filtering, are implemented in the programmable access device, and service functions, such as message interpretation, signaling, admission control, and policy invocation, are implemented in the external processor. This distribution of functionality results in numerous advantages, including improved scalability, flexibility, extensibility, interoperability, security, and service provisioning. (Specification, page 5, line 3 - page 6, line 7, FIGs. 2-4, 7A and 7B, claims 1, 20, 21 and 40)

If filtering functionality of the programmable access device (PAD) 40 detects packet flows for which services, additional to typical services afforded by the configuration to incoming and outgoing packets are appropriate, the programmable access device 40 passes appropriate messages to the external processor 42 for service processing via a Message, Control, and Reporting Interface (MCRI) 58, which can be accessed via an Application Programming Interface (API) on the programmable access device 40 and external processor 42. Distributing functionality between access router 44, programmable access device 40 and external processor 42 in this manner gives the service provider (or even third parties) the freedom to extend and modify existing services, create new services, or add more processing power to external processor 42 without adversely affecting the forwarding performance of the programmable access device 40 and the routing performance or functionality of access router 44.

To implement a desired functionality for programmable access device 40 and external processor 42, the service provider (or even a customer or third party) can define policy rules in the policy database 46 of one or more servers 48 (also referred to as a policy decision point (PDP)). Policy server 48 then makes policy decisions that control the functionality and operation of programmable access device 40 and external processors 42 by reference to the policy rules stored in policy database 46. Policy server 48 communicates policy decisions and associated configuration parameters for external processor 42 via a Service Policy Interface (SPI) 56, which can be accessed, for example, via an application program interface (API) on policy server 48 and external processor 42. Communication via Service Policy Interface 56 can employ any of a number of policy query protocols, including Common Open Policy Service (COPS) and Lightweight Directory Access Protocol (LDAP), which are respectively defined by Internet Engineering Task Force (IETF) RFCs 2748 and 2251. External processor 42 relays configuration parameters for programmable access device 40, if any, to programmable access device 40 via Message, Control, and Reporting Interface 58. (Specification, page 12, lines 4-31, FIGs. 2 and 4, claims 1, 2, 15, 17, 21, 22, 34, 36 and 40)

The functional modules of programmable access device 40 are logically arranged in incoming (e.g., from customer router 32) and outgoing (e.g., to customer router 32) traffic paths, with the incoming path including packet header filter 80, marker/policer 82, monitor(s) 84, forwarding table 86, and output buffers and scheduler 88. The outgoing path similarly includes packet header filter 90, forwarding table 86, monitor(s) 92, marker/shaper 94, and output buffers and scheduler 96. The functions of all of these functional modules can be independently configured or programmed by an external processor 42 through Message, Control, and Reporting Interface 58.

Incoming packets received from customer router 34 at the external interface of programmable access device 40 are first processed by packet header filter 80, which distinguishes between various message types using any one or a combination of the protocol type, Source Address (SA), Destination Address (DA), Type Of Service (TOS), Diffserv Codepoint (DSCP), Source Port (SP), Destination Port (DP), and other fields of a packet (e.g., layer 4 and higher layer fields such as the SYN, ACK, RST, and FIN TCP flags) upon which packet header filter 80 is configured to filter. In addition to filtering on layer-3 information, packet header filter 80 has the ability to identify higher layer (i.e., layer 4-7) message types or specific fields and forward those messages from/to external processor 42 based on the configured filter parameters. Thus, based upon its filter configuration and the fields of an incoming packet, packet header filter 80 directs the packet either to an external processor 42 via message interface 100 or to a specific marker/policer 82. Message interface 100 may also inject a packet specified by external processor 42 into either of packet header filters 80 and 90. (Specification, page 14, lines 7-32, FIGs. 2 and 3, claims 2-7, 13, 8-11, 28-31, 22-27 and 32)

After processing by packet header filter 80, incoming packets are processed by forwarding table 86. Forwarding table 86 maintains entries for each forwarding path, where each forwarding path is represented by packet flow attributes, such as DA, SA, TOS, PT, SP, DP, the incoming port, and the corresponding output port to which programmable access device 40 forwards the packet through the access network toward access router 44. Utilizing these forwarding table entries, forwarding table 86 forwards packets to the appropriate output ports and passes the packets to output buffers and scheduler 88. Output buffers and scheduler 88 buffer packets ready for transmission over communication network 30 and schedule the transmission of such packets. (Specification, page 15, line 29 - page 16, line 9, FIGs. 2 and 3, claims 10 and 29)

The outgoing path through programmable access device 40 is similar to the incoming path, except for the inclusion of marker/shaper 94 in lieu of marker/policer 82. Marker/shaper 94 discards nonconforming packets, sends marked packets to appropriate output buffers for the various queues serving different QoS classes for individual flows within output buffers and scheduler 96 to control the delay, jitter and loss of an outgoing packet flow, or simply counts non-conforming packets. (Specification, page 16, lines 24-30, FIGs. 2 and 3, claims 7 and 27)

The external processor 42 performs three types of processing: invoking policy services, signaling to setup and teardown access network connections, and configuring one or more associated programmable access devices 40. To coordinate these different processing functions, external processor 42 contains one or more service controllers 120, which each may control these three functions for a respective type of service. For example, service controllers 120 may include any or all of a Conference Call Service Controller (CCSC), an E-Commerce Service Controller (ECSC), an IP Telephony Service Controller (IPTELSC), a Reserved Bandwidth Service Controller (RBSC), and a Multicast Service Controller (MSC). Each service controller may maintain a session table recording all of its active sessions with a programmable access device.

As further shown in FIG. 4, external processor 42 includes, for each associated programmable access device 40, a respective programmable access device controller 124. Under the direction of service controller(s) 120, each programmable access device controller 124 configures forwarding table 86, packet header filters 80 and 90, marker/policer 82, marker/shaper 94, monitors 84 and 92, and output buffers and schedulers 88 and 96 of the associated programmable access device 40 by invoking commands or scripts understood by control interface 104. External processor 42 also contains a respective message processor 122 for each associated programmable access device 40. Message processors 122 each communicate messages to and

from the message interface 100 of the associated programmable access device 40. Upon receipt of a message from a programmable access device 40, which is usually a message received from the customer router 32, a message processor 122 parses the message and informs the appropriate service controller (as determined by the type of service) of its contents. (Specification, page 18, lines 4-31, FIGs. 3 and 4, claim 39)

In response to receipt of a policy decision from policy server 48, service controller 120 may inject one or more packets into a traffic flow via message processor 122, configure a programmable access device 40 via programmable access device controller 124 or control signaling inside or outside communication network 30 via signaling controllers 128a and 128b. Signaling controllers 128 support signaling protocols (e.g., Resource ReSerVation Protocol RSVP, Label Distribution Protocol (LDP), Private Network-Network Interface (PNNI), frame relay or ATM User Network Interface (UNI), etc.) to setup or tear down a Virtual Connection (VC) or Label Switched Path (LSP) across the network. A VC or LSP setup by a signaling controller 128 may have a specified Quality of Service (QoS). (Specification, page 19, lines 22-31, FIGs. 2 and 4)

Reporting interface 102 sends reporting messages to reporting processor 126 of external processor 42. The reporting messages tabulated in Table II, shown on pages 24-25 of the specification, include messages providing information about monitored sessions, messages related to communication between programmable access device 40 and service controllers 120 of external processor 42, and messages containing statistics collected by monitors 84 and 92. For protocols such as TCP and SIP, programmable access device 40 implements a state machine for each active session. If a TCP state machine detects that a particular active TCP session has had a number of retransmissions in excess of an established retransmission threshold, reporting

interface 102 sends a message notifying message processor 122 of external processor 42 that the TCP retransmission threshold has been exceeded, thus indicating that the TCP session has failed. Reporting processor 126 similarly reports other session failures such as the expiration of an inactivity timer on certain IP protocol sessions, such as TCP and SIP. For other data flows (e.g., UDP sessions) that do not have associated state machines to ensure reliability, reporting interface 102 of programmable access device 40 sends "Activity Detected" reporting messages when activity is detected in the session.

The connection state between a programmable access device 40 and external processor 42 is indicated by keepalive messages that are periodically exchanged between each programmable access device 40 and the associated external processor 42. The absence of a keepalive message programmable access device 40 indicates the failure of programmable access device 40 itself. (Specification, page 22, line 32 - page 23, line 26, FIGs. 2, 3 and 4, claim 19)

Table II lists two exemplary reporting messages triggered by the monitoring performed by monitors 84 and 92. First, reporting interface 102 can provide general usage statistics on a per-customer basis. Service controllers 120 in external processor 42 can utilize this statistical information to measure conformance to SLAs and detect certain events of interest. Second, reporting interface 102 can specifically indicate in a reporting message that a customer's predefined traffic threshold has been exceeded. (Specification, page 24, lines 13-24, FIGs. 2, 3 and 4)

As shown in Table III on pages 25-26 of the specification, the control messages sent from programmable access device controller 124 to control interface 104 via Message, Control, and Reporting Interface 58 include a number of configuration messages that enable a programmable access device controller 124 to configure any of the filtering, marking, policing, monitoring,

buffering, scheduling, shaping and forwarding functional modules 80-96 of programmable access device 40 through control interface 104. In particular, output buffers and schedulers 88 and 96 can be configured to allocate a number of buffers or size of buffer per traffic class or traffic flow or to implement CBQ, WFQ, WRR or other buffer scheduling algorithms. Programmable access device controller 124 can also configure marker/shaper 94 to employ static or adaptive shaping algorithms and can configure marker/shaper 94 to implement shaping on a per traffic flow or per traffic class basis. Programmable access device controller 124 can further configure forwarding table 86 in response to a request by a service controller 120 in order to enable the service controller 120 to associate a data flow with an ATM SVC or a MPLS LSP.

In addition to general control messages utilized to configure functional modules 80-96, Message, Control, and Reporting Interface 58 also supports various control messages utilized to configure particular features of the functional modules of programmable access device 40. For example, packet header filters 80 and 90 can be configured to drop multicast packets from an unauthorized source, to admit or deny source routing for a data flow, or to admit only packets with specific source addresses. In addition, programmable access device controller 124 can update forwarding table 86 with SVC and LSP paths setup by a service controller 120 using a signaling controller 128. Reporting interface 102 can be configured via a "Set reporting flags" control message to enable or disable reporting of selected events by setting or resetting reporting flags corresponding to these events. Programmable access device 40 can also be configured via Message, Control, and Reporting Interface control messages to set the TCP retransmission notification threshold, inactivity timers, activity timers and traffic threshold. Finally, the processing resources of programmable access device 40 and output buffers and scheduler 88, 96 can be configured by an "Allocate Resource" control message sent via Message, Control, and

Reporting Interface 58 and control interface 104 to dynamically allocate resources, such as bandwidth, queues, and processing time slices, to a customer interface, a packet flow, a class, or a multicast group. The reporting messages sent from reporting processor 126 of external processor 42 to programmable access device 40 are generally limited to exchanging keepalive messages with reporting interface 102. The continued exchange of keepalive messages informs programmable access device 40 that the associated service controller 120 is operative. (Specification, page 25, line 14 - page 26, line 22, FIGs. 2, 3, 4, claims 12, 16, 18, 31, 35 and 37)

The reporting messages sent from reporting processor 126 of external processor 42 to programmable access device 40 are generally limited to exchanging keepalive messages with reporting interface 102. The continued exchange of keepalive messages informs programmable access device 40 that the associated service controller 120 is operative.

With reference to FIG. 7D, a time-space diagram illustrates exemplary network access system signaling to close a TCP connection in accordance with the claimed invention. In the example shown in FIG. 7D, the server application initiates closure of the TCP session by instructing its TCP agent to close the connection. Accordingly, the server's TCP agent sends a FIN segment, informing the client application that it will send no more data. In response to receipt of FIN segment, programmable access device 40 resets the TCP state machine for the connection to idle state 142 and passes the FIN segment to e-commerce service controller (ECSC) 120. E-commerce service controller 120 responds by deleting the TCP session from its active session table and by configuring programmable access device 40 to stop marking packets for this TCP session and to remove the session's inactivity timer and retransmission setting. Programmable access device 40 also forwards FIN segment to the client, which acknowledges receipt of the FIN segment with an ACK that is passed to the server by programmable access

device 40. The client application then commands its TCP agent to close the session. The client's TCP agent therefore sends a FIN message to the server's TCP agent via programmable access device 40. (Specification, page 37, lines 11-31, FIGs. 2, 3, 7A and 7D, claims 14, 26, 33 and 37)

With reference to FIG. 7F, a route between a customer and a server is disrupted by failure of a network link or node. This failure causes the TCP agent and the client to re-transmit the data until a threshold number of retransmissions is reached. The client's TCP agent then aborts the TCP connection. Subsequently, the inactivity timer for the TCP session in programmable access device 40 expires. In response to expiration of the inactivity timer, programmable access device 40 updates state machine 140 of the TCP session to idle state 142 and reports the TCP session timeout error to e-commerce service controller 120. E-commerce service controller 120 responds to the report of the timeout error by deleting the TCP session from its active session table and instructs programmable access device 40 to stop marking the packets for the TCP session and to delete the configuration for this TCP session. Programmable access device 40 then deletes the state machine for the TCP session. (Specification, page 38, line 25 - page 39, line 3, FIGs. 2, 3, 4, 7A and 7F, claims 19 and 38)

In summary, a distributed network access system consistent with features of the claimed invention replaces a monolithic edge router with a programmable access device containing at least filtering and forwarding functionality, an external processor having one or more service-specific controllers that implement policy-based control of the programmable access device, and an access router that performs basic routing. This distributed architecture has numerous benefits over conventional monolithic router architectures, including scalability flexibility, extensibility, interoperability, security, and service provisioning. (Specification, page 50, line 29 - page 51, line 5, claim 38)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-4, 6-9, 11-15, 20-24, 26-29, 30-34 and 39-40 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* (U.S. 6,434,618) in view of *Bhattacharya et al.* (U.S. 6,587,466).

Whether claims 5 and 25 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* in view of *Bhattacharya et al.* and further in view of *Haas* (US 5,115,432).

Whether claims 16, 18, 35 and 37 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* in view of *Bhattacharya et al.* and further in view of *Feldmen et al.* (US 6,055,561).

Whether claims 17 and 36 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* in view of *Bhattacharya et al.* and further in view of *Sauter* (US 5,537,546).

Whether claims 19 and 38 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* in view of *Bhattacharya et al.* and further in view of *Grant et al.* (US 5,027,269).

Whether claims 10 and 29 are obvious under 35 U.S.C. § 103(a) based on *Cohen et al.* in view of *Bhattacharya et al.* and further in view of *Gai et al.* (US 6,651,096).

VII. ARGUMENT

A. CLAIMS 1-4, 6-9, 11-15, 20-24, 26-29, 30-34 and 39-40 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AL.*

The initial burden of establishing a *prima facie* basis to deny patentability to a claimed invention under any statutory provision always rests upon the Examiner. *In re Mayne*, 104 F.3d 1339, 41 USPQ2d 1451 (Fed .Cir. 1997); *In re Deuel*, 51 F.3d 1552, 34 USPQ2d 1210 (Fed. Cir. 1995); *In re Bell*, 991 F.2d 781, 26 USPQ2d 1529 (Fed. Cir. 1993); *In re Oetiker*, 977 F.2d 1443,

24 USPQ2d 1443 (Fed. Cir. 1992). In rejecting a claim under 35 U.S.C. § 103, the Examiner is required to provide a factual basis to support the obviousness conclusion. *In re Warner*, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 357 F.2d 385, 148 USPQ 721 (CCPA 1966); *In re Freed*, 425 F.2d 785, 165 USPQ 570 (CCPA 1970).

1. **Independent claims 1, 21 and 40**

Independent claim 1 recites, "**receiving a control message from the external processor, by the programmable access device, to establish a configuration of the programmable access device.**" Independent claim 21 recites, "an **external processor** that transmits **a control message specifying a configuration**" and "a programmable access device that receives messages from a first network external to the network access system via a first network interface, and that, **responsive to the control message, establishes the configuration specified by the control message.**" Independent claim 40 recites "an **external processor configured to receive, from the programmable access device, a first subset of the input messages and to transmit a control message to the programmable access device specifying a configuration to control the selection of the first subset.**"

In the Advisory Action, the Examiner adopts yet another interpretation of the claimed invention, in which the dispatcher process 402 (which has been previously equated to the claimed "external processor") of the *Cohen et al.* system is now used to satisfy the claimed "programmable access device," and one of the gateway programs 404-406 as the claimed external processor. Under this construction, the Examiner concludes that "Bhattacharya's idea of off loading the decision makers to an external device meaning the more complex resource intensive programs running, the gateway programs are taught to be moved externally to the dispatcher to reduce the complexity of the gateway router, and it would allow more than one gateway router to

be able to use the same decision maker." This reasoning is faulty, and at best contradictory, for the following reasons. First, the Examiner's proposed modification to the *Cohen et al.* system would change the principle operation and intended purpose of the programmable network element 400. Second, the teachings of *Bhattacharya et al.* would not motivate one of ordinary skill in the art to modify the *Cohen et al.* system, as the functions of the gateway programs are not what is suggested to be "off loaded."

The essence of the programmable network element 400 of the *Cohen et al.* system is defined by the gateway programs 404-407, which are the intelligence of the network element 400. The suggestion to remove these gateway programs 404-407 effectively redefines the function and purpose of the programmable network element 400, thereby changing the principle of operation of the *Cohen et al.* system.

Cohen et al. (see e.g., Abstract, col. 2: 5-16, 37-59 and col. 4: 7-14) provides a programmable network element 400 that operates on packet traffic flowing through the element in accordance with a gateway program 404, 405, 406 which is dynamically uploaded into the network element or unloaded from it via a mechanism separate from the actual packet traffic as the element operates. The programmable network element 400 can simultaneously operate on plural packet flows with different or the same programs being applied to each flow. A dispatcher 402 provides a packet filter 403 with a set of rules provided by one or more of the dynamically loaded and invoked programs. These rules define, for each program, the characteristics of those packets flowing through the network element that are to be operated upon in some manner. A packet that flows from the network through the filter and satisfies one or more of such rules is sent by the packet filter to the dispatcher 402. The dispatcher, in accordance with one of the programs, either sends the packet to the program for manipulation by the program itself, or

manipulates the packet itself in a manner instructed by the program. The processed packet is sent back through the filter to the network for routing to its destination. *Cohen et al.*'s programmable gateway 400 is embodied as a number of processes running on a Linux OS (col. 4: 7-14 and FIG. 4). In FIG. 4, the processes below dotted line 401 within gateway 400 represent processes within the Linux kernel.

The gateway programs 404-406 perform the functions that provide the network element 400 its programmability. These functions include encrypting or decrypting the payload of the packet, translating the destination address of the packet to an alternate address, as well as other functionalities associated with the program (col. 3: 49-52). Therefore, to remove these functions from the programmable gateway 400 would in fact eliminate its programmability, thereby changing the principle of operation of the gateway 400. Not surprisingly, the programs 404-406 are referred to as "gateway" programs, and thus, are not intended to be "off loaded." Otherwise, the gateway 400 would resemble other prior art routers; however, the *Cohen et al.* system seeks to provide a programmable router, as explained in Background section of the reference (col. 1: 30-35). If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). MPEP § 2143.01 Moreover, if a proposed modification would render the prior art being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

Second, in line with the above reasoning, the reliance on *Bhattacharya et al.*'s teaching is misguided, and would not suggest to one of ordinary skill to remove the gateway programs 404-

407 externally from the programmable network element 400. The Examiner relies on the following teaching within *Bhattacharya et al.* (col. 12: 8-12):

> ... Combined Policy-matching Engine may be located in an external policy server and policy decisions may be outsourced to this device, ...

Based on the teaching that policy decisions can be outsourced, the Examiner surmises on his own that other functions can be outsourced. There is no factual basis within *Bhattacharya et al.* to "off-load" any function other than a policy-matching function. *Bhattacharya et al.* is strictly concerned with approaches for constructing and using a search tree structure to accomplish policy based service differentiation.

Moreover, the Examiner's motivation for modifying the *Cohen et al.* system is suspect. The Examiner asserts in the Advisory Action that "the gateway programs are taught to be moved externally to the dispatcher to reduce the complexity of the gateway router." On the contrary, the off-loading of the gateway programs (even assuming it were technically possible) would introduce greater complexity.

Cohen et al. discloses the dispatcher process and the gateway programs forwarding packets back and forth between each other within the programmable network element without the need for a physical interface. Assuming that the dispatcher process and the gateway processes reside in physically separate devices, a physical interface would be required between these devices, entailing an increase in complexity. According to the Examiner's proposed modification, the communication between these devices would be over a network connection (per the *Bhattacharya* architecture). In addition to the circuitry, more logic is required to control the physical interface to exchange the packets. Forwarding packets back and forth between multiple devices would also result in high latency, which is an undesirable trait in computer networking.

Furthermore, this network-based solution would require more network resources (e.g., bandwidth).

Additionally, as noted above, the gateway programs 404, 405, 406 are **dynamically loaded and unloaded** from the programmable network element 400 **via local or remote program injectors and spawned by a local process called the admission daemon** (FIG. 1). Thus, from a technical standpoint, it would be infeasible to provide an external device for dynamic loading of the gateway programs, when local control is expressly desired.

A conclusion of obviousness is not compelled by the fact that the prior art could be modified so as to result in the combination defined by the claims; obviousness turns on whether the prior art suggests the desirability of the modification. The requisite motivation to establish a *prima facie* case of obviousness cannot be established by undercutting the expressed objectives of an applied reference. See *In re Fritch*, 972 F.2d 1260, 23 USPQ2d 1780 (Fed. Cir. 1992); *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984); *In re Schulpen*, 390 F.2d 1009, 157 USPQ 52 (CCPA 1968).

Accordingly, Appellants respectfully request that the rejection of claims 1-4, 6-9, 11-15, 20-24, 26-29, 30-34 and 39-40 be reversed.

2. Claims 2 and 22

For example, claim 2 recites "receiving a control message comprises receiving a filter control message to **establish a configuration of a packet header filter** in the programmable access device." For these features, the Examiner refers to col. 5: 20-25 and col. 5:66-col. 6:9 of *Cohen et al.* These passages state the following (Emphasis Added):

Col. 5: 20-25:

> Admission daemon 410 starts the execution of both locally injected and remotely injected gateway programs. Each gateway program 404, 405 and 406 is registered with the dispatcher process 402 by admission daemon 410, which also informs the dispatcher process 402 of the privilege level of the program.

> Col. 5: 66-col. 6:9:

> Further reduction in the size of messages which are transferred are achieved by certain gateway programs that instruct the dispatcher process itself to perform specific functionalities rather than having these same functionalities performed within a gateway program. For example, **packets can be filtered in accordance with whether they contain a specific flag**, such as the SYN flag, in the packet header. This flag, as is well known, marks a packet a being part of a TCP connection establishment protocol rather than a data packet for a particular connection.

These cited passages provide a general disclosure of a how a packet can be filtered by using a SYN flag. This falls short of configuring the packet filter 403. *Cohen et al.* explains, per col. 4: 11-15, that the dispatcher process 402 uses the packet filter process 403 in the Linux kernel to obtain packets requested by any of the gateway programs 404, 405 and 406. The dispatcher process 402 is the only process which interacts with the packet filter process 403. The packet filter 403 does not provide any capability to have its configuration controlled by the dispatcher process 402 or any other process, much less in the manner claimed.

3. Claims 4 and 24

For example, claim 4 recites "receiving a control message comprises **receiving a monitor control message to establish a configuration of a monitor in the programmable access device**." For a supposed teaching of the above features, the Examiner refers (Final Office Action, page 5) to col. 10: 25-32. This passages state the following (Emphasis Added):

> Further functions and data structures can be used which are related to the ability of the dispatcher process to carry out the processing of packets in accordance with instructions from the gateway programs. These functions and data structures are needed in cases where **a gateway program wishes to receive occasional**

feedback from the dispatcher process about the current state of the packet flow through the dispatcher process.

This passage provides no disclosure of a capability to "establish a **configuration** of a monitor", but merely suggests a mechanism for the dispatcher process to provide state of the packet flow to the gateway program.

B. CLAIMS 5 AND 25 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AT.* AND FURTHER IN VIEW OF *HAAS*

The obviousness rejection of claims 5 and 25 should also be reversed. Dependent claim 5, for example, recites "wherein receiving a monitor control message comprises **receiving a control message to establish a threshold number of allowed retransmissions**." *Haas* is applied merely for the disclosure of a retransmission policy and does not fill the gaps of *Cohen et al.* and *Bhattacharya et al.* mentioned above in Section VII. A.1, in regard to their independent claims.

Additionally, *Haas* discusses in generalities the retransmission policies can be changed (col. 7: 49-61). There is no discussion of the specific use of a threshold, much less "**a threshold number of allowed retransmissions.**"

C. CLAIMS 16, 18, 35 AND 37 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AT.* AND FURTHER IN VIEW OF *FELDMEN ET AL.*

With respect to the obviousness rejection of claims 16, 18, 35 and 37, this rejection should also be reversed. For example, dependent claim 16 recites "exchanging keepalive messages between the external processor and the programmable access device." *Feldmen et al.* does not fill in the gaps of *Cohen et al.* and *Bhattacharya et al.* (as explained in Section VII. A.1),

and is applied for a supposed teaching of exchanging keepalive and acknowledgement messages between the external processor and the programmable access device.

Additionally, the Examiner misapplies *Feldmen et al.* The cited passage, col. 9: 65-col. 10: 11, describes a keep-alive mechanism between neighboring routing nodes or endpoints. Based on this teaching, one of ordinary skill in the art would not be motivated to use keep-alive messages within the same routing environment (formed by gateway programs and the dispatcher of the *Cohen et al.* system).

Furthermore, to the extent the Examiner is relying on common knowledge, Appellants note that the Administrative Procedure Act requires the Patent Office to articulate and place on the record the "common knowledge" used to negate patentability. *In re Sang Su Lee*, No. 00-1158 (Fed. Cir., Jan. 18, 2002); *In re Zurko*, No. 96-1285 (Fed. Cir., Aug. 2, 2001).

D. CLAIMS 17 AND 36 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AT.* AND FURTHER IN VIEW OF *SAUTER*.

As for the obviousness rejection of claims 17 and 36, this rejection is unsustainable as the addition of *Sauter* to the combination of *Cohen et al.* and *Bhattacharya et al.* does not satisfy the claimed features (as argued above in Section VII. A.1). For example, dependent claim 17 recites "wherein receiving a control message comprises accessing a control processor on the external processor via an application programming interface." *Sauter* is relied upon for a supposed teaching of transmitting a control message comprises accessing a control processor on the external processor via an application programming interface.

E. CLAIMS 19 AND 38 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AT.* AND FURTHER IN VIEW OF *GRANT ET AL.*

As for claims 19 and 38, the obviousness rejection over the combination of *Cohen et al.* and *Bhattacharya et al.* in further view of *Grant et al.* should be reversed. For example, dependent claim 19 recites "communicating a state of a session from the programmable access device to the external processor in response to failure of **a service controller servicing the session in the external processor**." *Grant et al.* is applied for a supposed teaching of failure detection in a system where data is lost.

Even assuming *Grant et al.* is properly applied for the feature of detection of a failure, the Examiner has ignored the other features of "**a service controller servicing the session in the external processor**." This feature is not disclosed by the combination of *Cohen et al.* and *Bhattacharya et al.*

F. CLAIMS 10 AND 29 ARE NOT RENDERED OBVIOUS OVER *COHEN ET AL.* IN VIEW OF *BHATTACHARYA ET AT.* AND FURTHER IN VIEW OF *GAI ET AL.*

The obvious rejection of claims 10 and 29 over *Cohen et al.* and *Bhattacharya et al.* in view of *Gai et al.* is likewise unsustainable, as *Gai et al.* fails to fill in the gaps of *Cohen et al.* and *Bhattacharya et al.* (as argued in Section VII. A.1). For example, dependent claim 10 recites "receiving a control message comprises receiving a control message to establish a configuration of a scheduler and one or more associated output buffers in the programmable access device." *Gai et al.* is relied upon for a supposed disclosure of a system for controlling the configuration of an access device that includes making configuration changes to a scheduler and has one or more output queues.

The Examiner's conclusion of obviousness does not explain how and why one of ordinary skill in the art would want to alter the dispatcher of the *Cohen et al.* to include a scheduler. The Examiner is required to explain how and why one having ordinary skill in the art would have been led to modify an applied reference to arrive at the claimed invention. *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 5 USPQ2d 1434 (Fed. Cir. 1988). In establishing the requisite motivation, it has been consistently held that both the suggestion and the reasonable expectation of success must stem from the prior art itself, as a whole. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991); *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Dow Chemical Co.*, 837 F.2d 469, 5 USPQ2d 1529 (Fed. Cir. 1988). None of these requirements have been met for these claims.

VIII. CONCLUSION AND PRAYER FOR RELIEF

For the foregoing reasons, Appellants request the Honorable Board to reverse each of the Examiner's rejections. Appellants note that this is a second Appeal in the case, and yet, the Examiner has not settled on his search or his interpretation of the resulting references.

Respectfully Submitted,

DITTHAVONG & MORI, P.C.

11/14/2006
Date

For Phouphanomketh Ditthavong
Attorney for Appellant(s)
Reg. No. 44658

NAME: SANGWON KIM
Reg. No: 54,224

10507 Braddock Rd, Suite A
Fairfax, VA 22032
Tel. 703-425-8516
Fax. 703-425-8518

IX. CLAIMS APPENDIX

1. (Previously Presented) A method of communication in a network access system including an external processor and a programmable access device, said method comprising:

receiving a control message from the external processor, by the programmable access device, to establish a configuration of the programmable access device;

receiving, by the programmable access device, messages from a first network external to the network access system via a first network interface;

communicating a first subset of the received messages from the programmable access device to the external processor for service processing in accordance with the configuration; and

routing a second subset of the received messages not communicated to the external processor from the network access system via a second network interface different from the first network interface to a second network external to the network access system, wherein the second network is different from the first network.

2. (Previously Presented) The method of Claim 1, wherein:

receiving a control message comprises receiving a filter control message to establish a configuration of a packet header filter in the programmable access device; and

communicating messages comprises communicating network messages filtered from a packet flow by the packet header filter of the programmable access device.

3. (Original) The method of Claim 2, and further comprising limiting communication of network messages from the programmable access device to the external processor by sending the programmable access device a message setting message interface flags in the programmable access device.

4. (Previously Presented) The method of Claim 1, wherein:

receiving a control message comprises receiving a monitor control message to establish a configuration of a monitor in the programmable access device; and

communicating messages comprises communicating reporting messages from the programmable access device to the external processor in response to the configuration of the monitor.

5. (Previously Presented) The method of Claim 4, wherein receiving a monitor control message comprises receiving a control message to establish a threshold number of allowed retransmissions.

6. (Previously Presented) The method of Claim 4, wherein receiving a monitor control message comprises receiving a threshold activity level.

7. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises receiving a policer control message to establish a configuration of a policer in the programmable access device.

8. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises receiving a forwarding table control message to establish a configuration of a forwarding table in the programmable access device.

9. (Original) The method of Claim 8, wherein establishing a configuration of a forwarding table comprises establishing a new forwarding table in the programmable access device.

10. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises receiving a control message to establish a configuration of a scheduler and one or more associated output buffers in the programmable access device.

11. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises receiving a shaper control message to establish a configuration of a shaper in the programmable access device.

12. (Previously Presented) The method of Claim 1, wherein:

receiving a control message from the external processor, to the programmable access device, to establish a configuration of the programmable access device comprises receiving a control message specifying a source from which packets are not to be accepted; and

the method further comprises dropping packets from the specified source by the programmable access device.

13. (Original) The method of Claim 1, and further comprising in response to service processing by the external processor, injecting a packet from the external processor into packet flow through the programmable access device.

14. (Previously Presented) The method of Claim 1, wherein

receiving a control message from the external processor, to the programmable access device, to establish a configuration of the programmable access device comprises receiving a session deletion control message; and

the method further comprises the programmable access device deleting a session specified by the session deletion control message.

15. (Original) The method of Claim 1, and further comprising the external processor signaling network hardware to establish a network connection in response to receipt of a message from the programmable access device.

16. (Original) The method of Claim 1, and further comprising exchanging keepalive messages between the external processor and the programmable access device.

17. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises accessing a control processor on the external processor via an application programming interface.

18. (Original) The method of Claim 1, and further comprising in response to said control message, sending an acknowledgement from said programmable access device to said external processor.

19. (Original) The method of Claim 1, and further comprising communicating a state of a session from the programmable access device to the external processor in response to failure of a service controller servicing the session in the external processor.

20. (Previously Presented) The method of Claim 1, wherein receiving a control message comprises receiving a control message via an intermediate communication network.

21. (Previously Presented) A network access system, comprising:

an external processor that transmits a control message specifying a configuration; and

a programmable access device that receives messages from a first network external to the network access system via a first network interface, and that, responsive to the control message, establishes the configuration specified by the control message and

communicates a first subset of the received messages to the external processor for service processing in accordance with the configuration, and forwards a second subset of the received messages not communicated to the external processor for routing, via a second network interface different from the first network interface, to a second network external to the network access system, wherein the second network is different from the first network.

22. (Original) The network access system of Claim 21, wherein:

the programmable access device includes a packet header filter;

the control message comprises a filter control message that establishes a configuration of the packet header filter; and

the messages communicated by the programmable access device comprise network messages filtered from a packet flow by the packet header filter of the programmable access device.

23. (Original) The network access system of Claim 22, said external processor comprising means for limiting communication of network messages from the programmable access device to the external processor by sending the programmable access device a message setting message interface flags in the programmable access device.

24. (Original) The network access system of Claim 21, wherein:

the programmable access device comprises a monitor for network traffic;

the control message comprises a monitor control message that specifies a configuration of the monitor; and

the messages communicated by the programmable access device comprise reporting messages in accordance with the configuration.

25. (Original) The network access system of Claim 24, wherein the control message specifies a threshold number of allowed retransmissions.

26. (Original) The network access system of Claim 24, wherein the monitor control message specifies a threshold activity level.

27. (Original) The network access system of Claim 21, wherein:

the programmable access device comprises a policer, and

the control message comprises a policer control message that specifies a configuration of the policer.

28. (Original) The network access system of Claim 21, wherein the control message comprises a forwarding table control message that specifies a configuration for a forwarding table.

29. (Original) The network access system of Claim 21, wherein:

the programmable access device comprises one or more output buffers for outgoing packets and an associated scheduler; and

the control message specifies a configuration of the scheduler and the one or more output buffers.

30. (Original) The network access system of Claim 21, wherein:

the programmable access device comprises a shaper; and

the control message comprises a shaper control that specifies a configuration of the shaper.

31. (Original) The network access system of Claim 21, wherein:

the control message specifies a source from which packets are not to be accepted; and

the programmable access device comprises means for dropping packets from the specified source.

32. (Original) The network access system of Claim 21, said external processor comprising means, responsive to service processing by the external processor, for injecting a packet into packet flow through the programmable access device.

33. (Original) The network access system of Claim 21, wherein

the control message comprises a session deletion control message; and

the programmable access device comprises means for deleting a session specified by the session deletion control message.

34. (Original) The network access system of Claim 21, wherein the external processor comprises a signaling processor that signals network hardware to establish a network connection in response to a message received from the programmable access device.

35. (Original) The network access system of Claim 21, said external processor and said programmable access device each comprising means for exchanging keepalive messages.

36. (Original) The network access system of Claim 21, wherein the external processor comprises a control processor that outputs said control message and an application programming interface through which said control processor is accessed.

37. (Original) The network access system of Claim 21, said programmable access device comprising means, responsive to said control message, for sending an acknowledgement to said external processor.

38. (Original) The network access system of Claim 21, wherein:

the external processor comprises a plurality of service controllers that provide service processing; and

the programmable access device comprises means for communicating a state of a session to the external processor in response to failure of a service controller servicing the session.

39. (Original) The network access system of Claim 21, and further comprising a network coupling the external processor and the programmable access device.

40. (Previously Presented) A distributed router comprising:

a first network interface through which packets are communicated with a first network;

a second network interface different from the first network interface through which packets are communicated with a second network different from the first network;

a programmable access device configured to input messages from the first network via the first network interface; and

an external processor configured to receive, from the programmable access device, a first subset of the input messages and to transmit a control message to the programmable access device specifying a configuration to control the selection of the first subset,

wherein the programmable access device forwards a second subset of the input messages not received by the external processor for routing via the second network interface to the second network.

X. EVIDENCE APPENDIX

Appellants are unaware of any evidence that is required to be submitted in the present Evidence Appendix.

XI. RELATED PROCEEDINGS APPENDIX

Appellants are unaware of any related proceedings that are required to be submitted in the present Related Proceedings Appendix.